FIG. 1

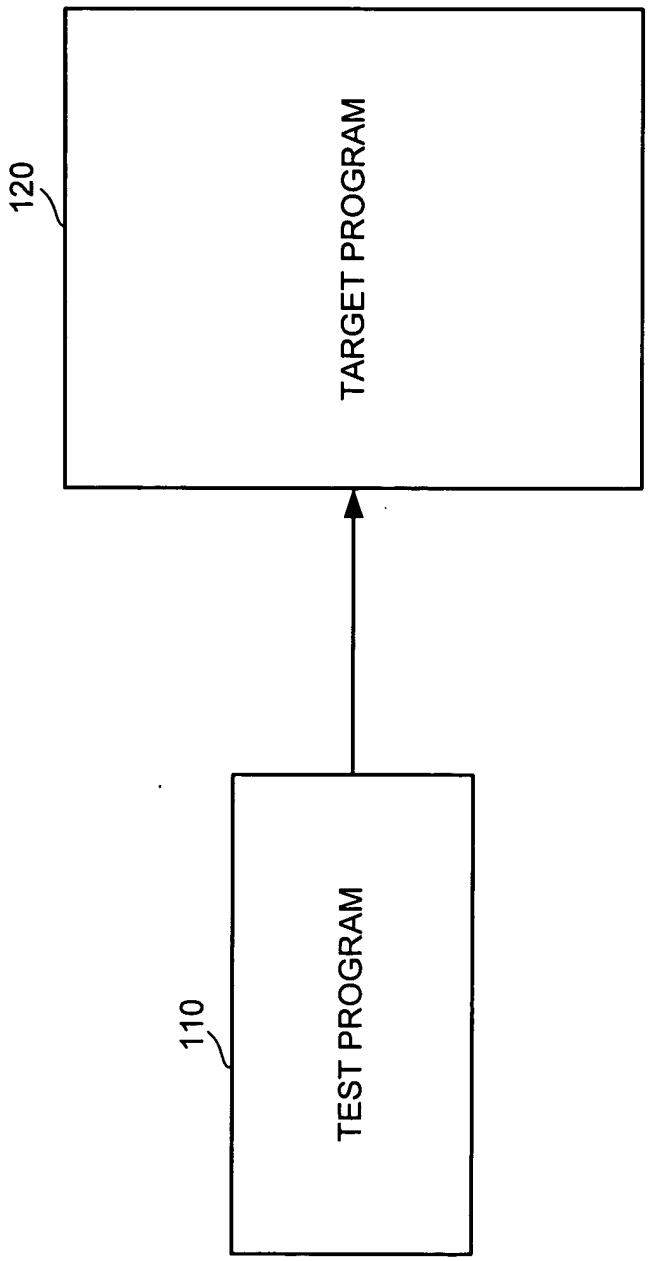


FIG. 2

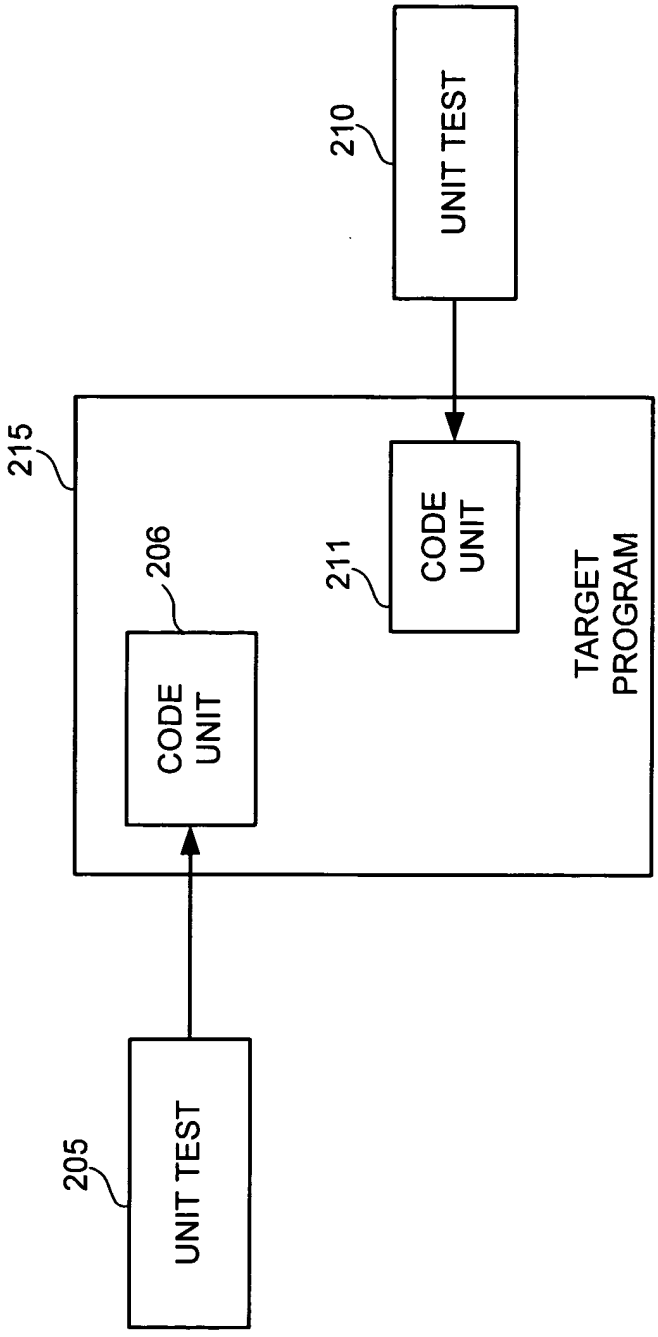

UNIT TEST
210
215
CODE UNIT
211
206
CODE UNIT
TARGET PROGRAM
UNIT TEST
205

FIG. 3

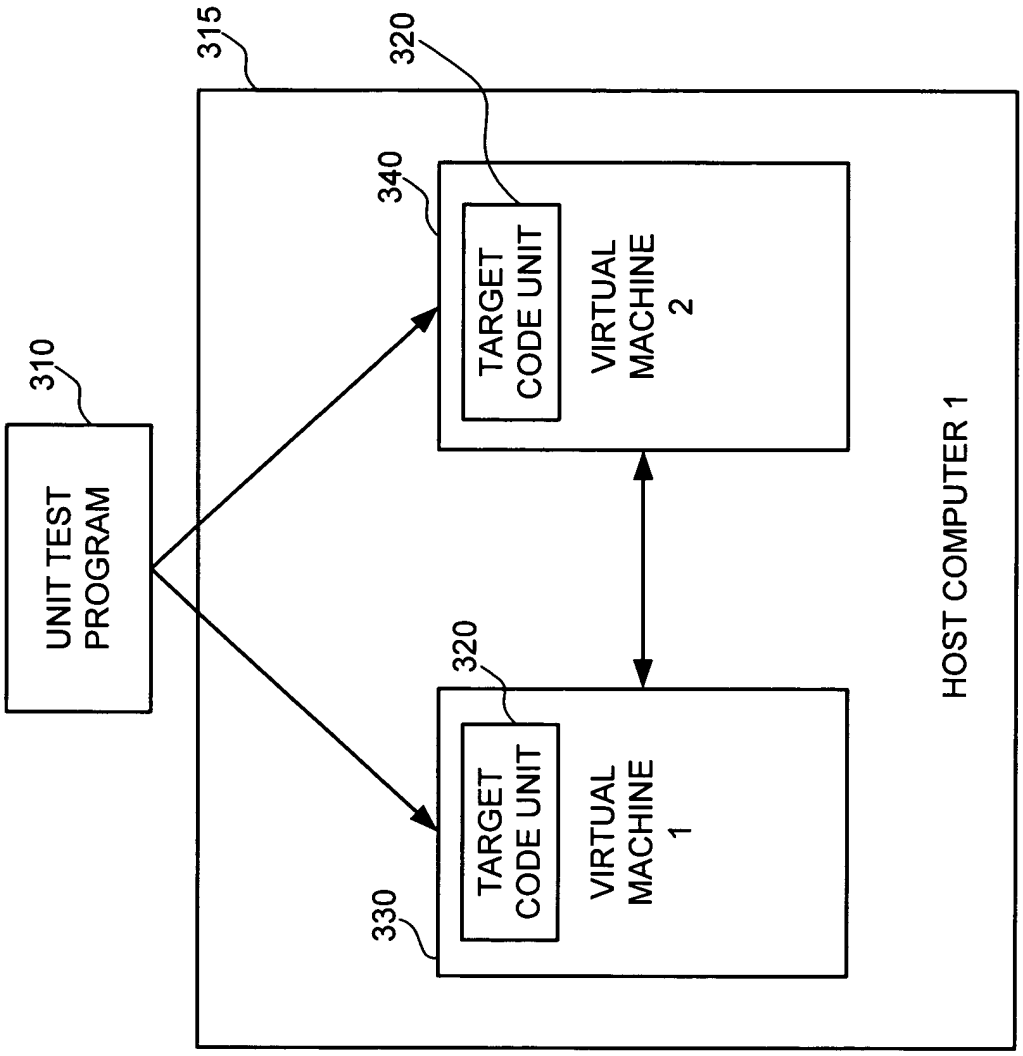
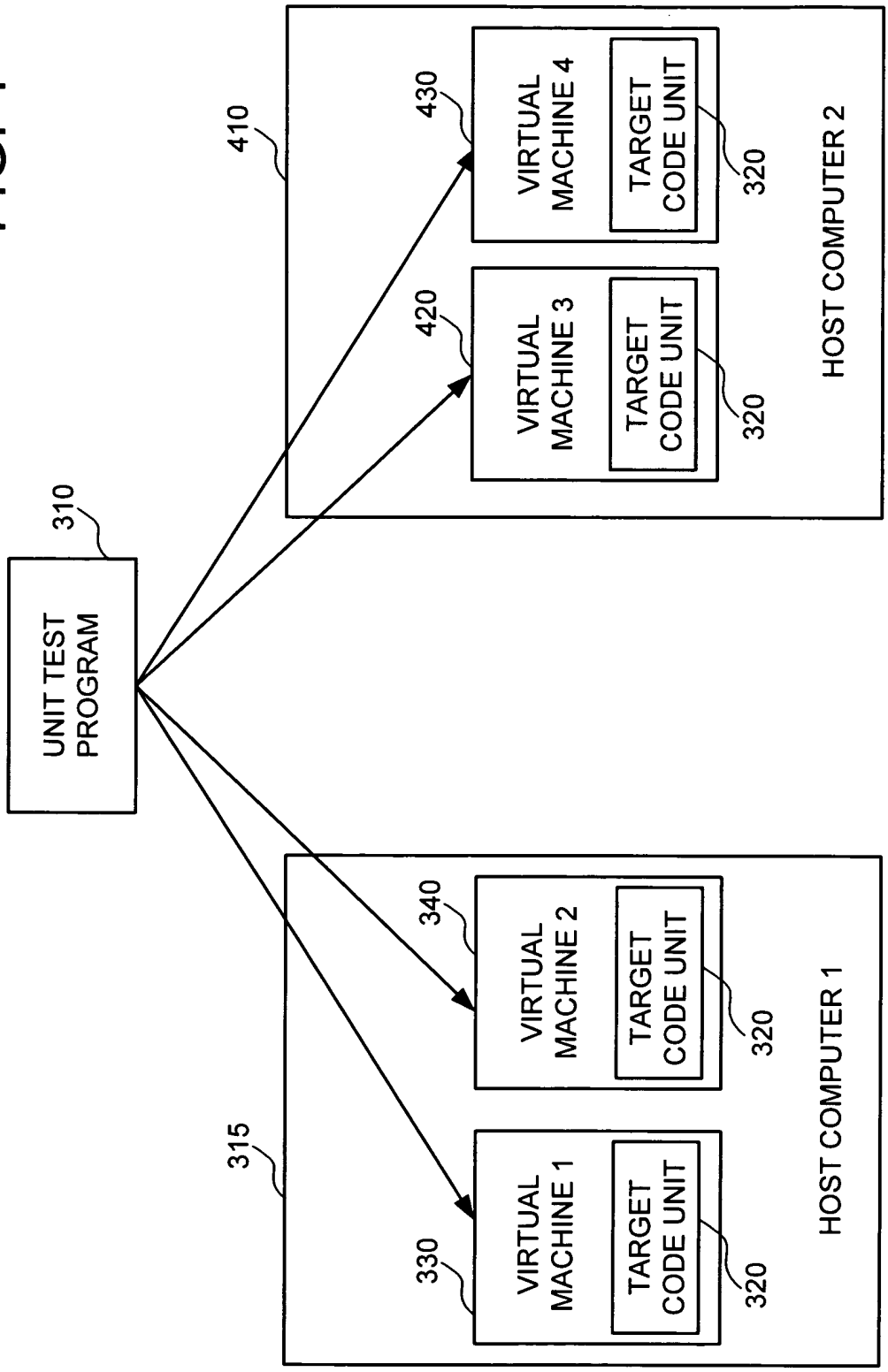

315
320
340
TARGET CODE UNIT
VIRTUAL MACHINE 2
310
UNIT TEST PROGRAM
320
TARGET CODE UNIT
VIRTUAL MACHINE 1
330
HOST COMPUTER 1

FIG. 4


UNIT TEST PROGRAM
310
HOST COMPUTER 1
315
VIRTUAL MACHINE 1
330
TARGET CODE UNIT
320
VIRTUAL MACHINE 2
340
TARGET CODE UNIT
320
HOST COMPUTER 2
410
VIRTUAL MACHINE 3
420
TARGET CODE UNIT
320
VIRTUAL MACHINE 4
430
TARGET CODE UNIT
320

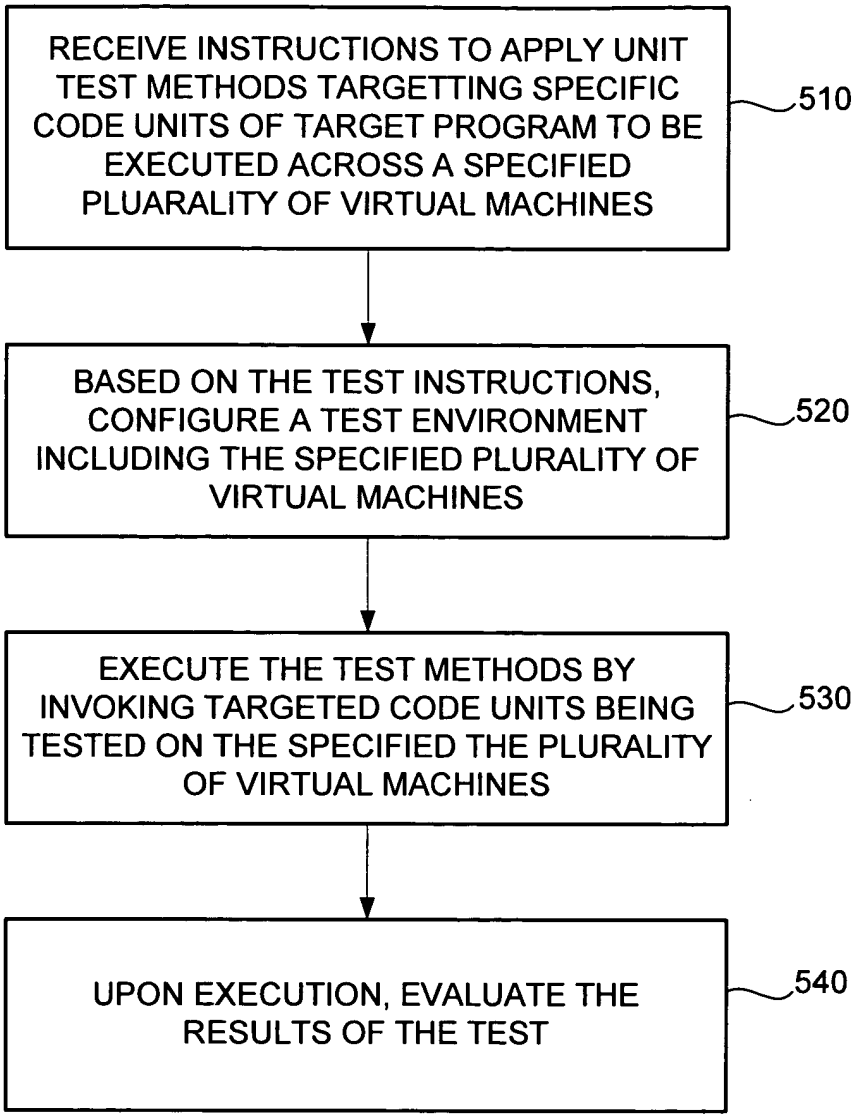

RECEIVE INSTRUCTIONS TO APPLY UNIT TEST METHODS TARGETTING SPECIFIC CODE UNITS OF TARGET PROGRAM TO BE EXECUTED ACROSS A SPECIFIED PLUARALITY OF VIRTUAL MACHINES
510
BASED ON THE TEST INSTRUCTIONS, CONFIGURE A TEST ENVIRONMENT INCLUDING THE SPECIFIED PLURALITY OF VIRTUAL MACHINES
520
EXECUTE THE TEST METHODS BY INVOKING TARGETED CODE UNITS BEING TESTED ON THE SPECIFIED THE PLURALITY OF VIRTUAL MACHINES
530
UPON EXECUTION, EVALUATE THE RESULTS OF THE TEST
540

FIG. 5

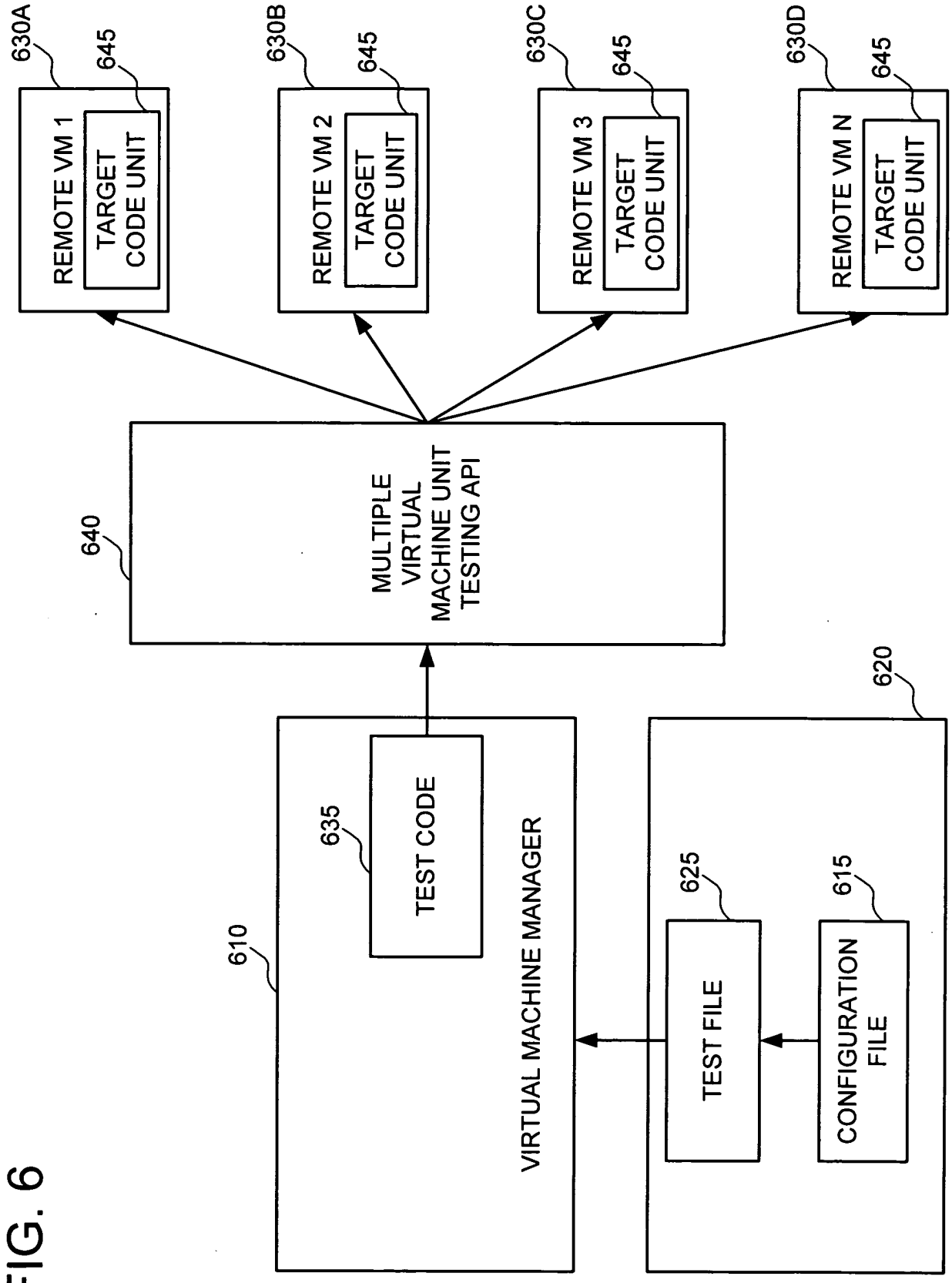

630A
645
REMOTE VM 1
TARGET CODE UNIT
630B
645
REMOTE VM 2
TARGET CODE UNIT
630C
645
REMOTE VM 3
TARGET CODE UNIT
630D
645
REMOTE VM N
TARGET CODE UNIT
640
MULTIPLE VIRTUAL MACHINE UNIT TESTING API
620
635
TEST CODE
610
VIRTUAL MACHINE MANAGER
625
TEST FILE
615
CONFIGURATION FILE

FIG. 6

```
// This config file is used to run Distributed Unit Tests
INCLUDE $JTESTS/dunit/impl/dunit.inc;
UNITTEST
   testClass - dunit.tests BasicTest
   testMethod - test RemotInvocation
# Runs all test* methods in OwnershipTest
   testClass = com.gemstone.gemfire.interal.cache.OwnershipTest
;
```

710

715

720

615

FIG. 7

```
package dunit.tests;
import com.gemstone.gemfire.*;
import dunit.*;
public class BasicTest extends DistributedTestCase {
 public void testRemoteInvocation()
    Host host = Host.getHost(0);                                    805
    VM vm1 - host - Host.getHost(0)                                 810
    VM vm2 = host.get VM(1)                                         815
    String name - this.getUniqueName();
    String value - "Hello";
820 vm1.invoke(this.getClass(), "remoteBind",                       821
        new Object[] { name, value });
825 vm2.invoke(this.getClass(), "remoteValidateBind",               826
        new Ojbect [] {name, value });
 }
 private static void remoteBind(String name, String s) {
    GemFireConnection conn = GemFireConnetionFactory.getInstance():  830
    Namespace ns = conn.getNamespace();
    ns.put(name, s);
 }
 private static void remoteValidateBind(String name, String expected)  835
 {
    GemFireConnection conn = GemFireConnectionFactory.getInstance();
    Namespace ns = conn.getNamespace();
    assertEquals(expected, ns.get());
 }
}
```

800

FIG. 8